

Docket No. RSW920030145US1

CLAIMS:

What is claimed is:

- 1 1. A system for analyzing thread deadlocks in a Java
2 virtual machine, comprising:
3 a thread analyzer tool, wherein the tool analyzes a
4 thread dump to automatically identify thread deadlocks;
5 wherein the tool identifies threads that are in a
6 self wait condition and threads that are in a circular
7 wait condition.
- 1 2. The system of claim 1, further comprising a user
2 interface that allows a user to specify criteria, wherein
3 the tool excludes threads that do not meet the criteria
4 from identification as deadlocked threads.
- 1 3. The system of claim 1, wherein the tool analyzes a
2 thread dump file in offline mode.
- 1 4. The system of claim 1, wherein the tool obtains a
2 thread dump from a running information processing system.
- 1 5. A method of analyzing thread deadlocks in a Java
2 virtual machine, comprising the steps of:
3 obtaining a thread dump of a Java virtual machine;
4 analyzing the thread dump to automatically identify
5 threads in a deadlock condition;
6 wherein threads in a circular wait condition and
7 threads in a self wait condition are identified.

Docket No. RSW920030145US1

1 6. The method of claim 5, further comprising a user
2 interface that allows a user to specify criteria, wherein
3 threads that do not meet the criteria are filtered.

1 7. The method of claim 6, wherein filtered threads are
2 excluded from identification as threads in a self wait
3 condition.

1 8. The method of claim 5, wherein the tool analyzes a
2 thread dump file in offline mode.

1 9. The method of claim 5, wherein a matrix is populated
2 with threads owning resources and threads waiting on
3 resources, and wherein the matrix is used to identify
4 threads in a circular wait condition.

1 10. The method of claim 5, wherein the step of analyzing
2 the thread dump to automatically identify threads in a
3 deadlock condition includes the steps of:

4 (a) identifying threads that own resources;

5 (b) identifying threads that are waiting on
6 resources;

7 comparing the results from steps (a) and (b) to
8 identify threads in a circular wait condition.

1 11. A method of analyzing thread deadlocks in a Java
2 virtual machine (JVM), comprising the steps of:

3 obtaining a thread dump file;

4 identifying waiting threads;

Docket No. RSW920030145US1

5 identifying locking threads; and
6 comparing waiting threads and locking threads to
7 identify threads in a self wait condition.

1 12. The method of claim 11, further comprising the step
2 of comparing waiting threads and locking threads to
3 identify threads in a circular wait condition.

1 13. The method of claim 11, wherein the step of
2 obtaining a thread dump file comprises obtaining a thread
3 dump from a live JVM.

1 14. The method of claim 11, wherein the step of
2 obtaining a thread dump file comprises opening an
3 existing thread dump file.

1 15. The method of claim 14, wherein the existing thread
2 dump file is analyzed in offline mode.

1 16. The method of claim 11, wherein a user interface
2 allows a user to choose rules, and wherein the rules are
3 used to exclude threads from being identified as in a
4 deadlock condition.

1 17. A system for analyzing thread deadlocks in a Java
2 virtual machine (JVM), comprising the steps of:
3 means for obtaining a thread dump file;
4 means for identifying waiting threads;
5 means for identifying locking threads; and

Docket No. RSW920030145US1

6 means for comparing waiting threads and locking
7 threads to identify threads in a self wait condition.

1 18. The system of claim 17, further comprising means for
2 comparing waiting threads and locking threads to identify
3 threads in a circular wait condition.

1 19. The system of claim 17, wherein thread deadlocks are
2 analyzed in offline mode.

1 20. A computer program product in a computer readable
2 medium for analyzing thread deadlocks in a Java virtual
3 machine, comprising:
4 first instructions for obtaining a thread dump file;
5 second instructions for identifying waiting threads;
6 third instructions for identifying locking threads;
7 and
8 fourth instructions for comparing waiting threads
9 and locking threads to identify threads in a self wait
10 condition.

1 21. The computer program product of claim 20, further
2 comprising fifth instructions for comparing waiting
3 threads and locking threads to identify threads in a
4 circular wait condition.

1 22. The computer program product of claim 20, wherein
2 thread deadlocks are analyzed in offline mode.